

Homework 12 - Red Neuronal

Claudia Lizeth Hernández Ramírez

17 de noviembre de 2021

1. Introducción

Estudia de manera sistemática el desempeño de la red neuronal en términos de su F-score para los diez dígitos en función de las tres probabilidades asignadas a la generación de los dígitos (ngb), variando a las tres en un experimento factorial adecuado.

2. Desarrollo

Trabajé con el código visto en clase [5], al cual se le agregaron algunas modificaciones para cumplir con el objetivo de la tarea. Antes que otra cosa, investigué sobre el **F-Score**, y me percaté de que hay tres fórmulas para obtenerlo [2] [1].

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (1)$$

$$Recall = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2)$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

Con esta información adecué las ecuaciones 1, 2 y 3 para mi código.

Listing 1: Segmento de código ecuación F-Score.

```
precision = diag(contadores) / colSums(contadores[,1:10])
recall <- diag(contadores) / rowSums(contadores)
fscore <- (2 * precision * recall) / (precision + recall)
```

También trabajé con varios ciclos **FOR** con los cuales variaba la probabilidad con la que mis cuadros **negro**, **gris** y **blanco** aparecerían. Trabajé con 6 replicas.

Cuadro 1: Probabilidades por grupo.

	Negro	Gris	Blanco
1	0.8	0.5	0.002
2	0.9	0.8	0.01
3	0.995	0.92	0.5

Listing 2: Segmento de código ciclos **FOR**.

```
replica = 1:6 #Replicaas
negro<-c(0.8, 0.9, 0.995) #probabilidad encender cuadritos negros
gris<-c(0.5, 0.8, 0.92) #probabilidad encender cuadritos grises
```

```

blanco<-c(0.002, 0.01, 0.5) #probabilidad encender cuadritos blancos

for(ne in negro){
  for(gr in gris){
    for(bl in blanco){
      for(reply in replica){
        modelos <- read.csv("digits.txt", sep=" ",
                           header=FALSE,
                           stringsAsFactors=F)

        modelos[modelos=='n'] <- ne
        modelos[modelos=='g'] <- gr
        modelos[modelos=='b'] <- bl
      }
    }
  }
}

```

Toda esta información la almacené en una **Data Frame**.

Listing 3: Segmento de código Data Frame.

```

eale = c(reply, ne, gr, bl, fscore)
datos = rbind(datos, eale)
names(datos) = c("Replica", "Negro", "Gris", "Blanco",
                 "0", "1", "2", "3", "4", "5", "6", "7", "8", "9")

```

Posteriormente realicé la gráfica 1 .

Listing 4: Segmento de código Gráfica.

```

gr = ggplot(dat.m, aes(x= combo, y= value, fill= combo)) +
  geom_violin(fill = "#D29BFD")
gr + geom_boxplot(fill = colorRampPalette(c("#FF1694", "#FC46AA", "#F25278",
                                           "#FC4C4E", "#FE7D6A", "#FC9483",
                                           "#FC94AF", "#F79AC0", "#FA86C4"))(27),
              width = 0.3, lwd = 0.3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        panel.background = element_rect(fill = "#CDCDCD")) +
  labs(x="combinacion de probabilidades", y= "f-score")

```

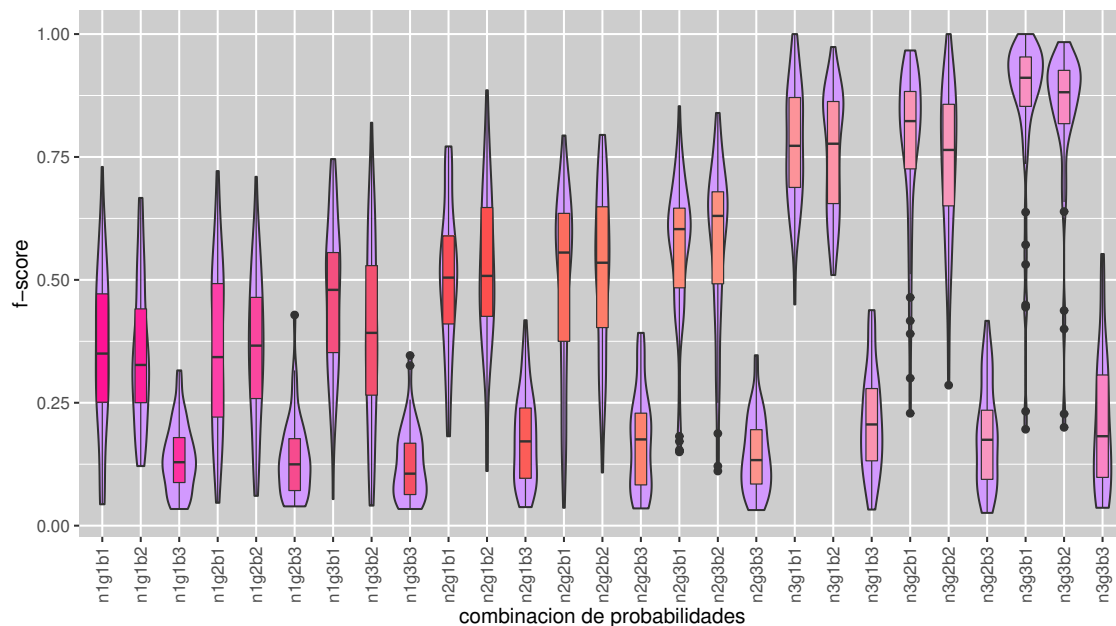


Figura 1: F-Score vs Probabilidad.

3. Estadística

Listing 5: Segmento de código Pruebas Estadísticas .

```
#estad\`isitica -
#con p menor a 0.05 se rechaza hipotesis nula H0
#H0: los datos proceden de una distribuci\`on normal
#H1: los datos no proceden de una distribuci\`on normal
tapply(dat.m$value, dat.m$combo, shapiro.test)

ala = dat.m%%
group_by(combo) %%
summarise(
  cantidad_de_participantes = n(),
  promedio = mean(value, na.rm = TRUE),
  desviacion_estandar = sd(value, na.rm = TRUE),
  varianza = sd(value, na.rm = TRUE)^2,
  mediana = median(value, na.rm = TRUE),
  rango_intercuartil = IQR(value, na.rm = TRUE)
)

kruskal.test(value ~ combo, data = dat.m)
```

Cuadro 2: Resultados obtenidos de prueba de normalidad de Shapiro.

Combinación	W value	P value	¿Se acepta H0?
n1g1b1	0.9780	0.3508	sí
n1g1b2	0.9550	0.0271	no
n1g1b3	0.9590	0.0584	sí
n1g2b1	0.9715	0.1818	sí
n1g2b2	0.9889	0.8635	sí
n1g2b3	0.8996	0.0003	no
n1g3b1	0.9809	0.4918	sí
n1g3b2	0.9809	0.4800	sí
n1g3b3	0.9210	0.0020	no
n2g1b1	0.9774	0.3292	sí
n2g1b3	0.9887	0.8533	sí
n2g1b3	0.9593	0.0600	sí
n2g2b1	0.9110	0.0003	no
n2g2b2	0.9664	0.0981	sí
n2g2b3	0.9382	0.0065	no
n2g3b1	0.9074	0.0002	no
n2g3b2	0.9184	0.0006	no
n2g3b3	0.9517	0.0373	no
n3g1b1	0.9822	0.5328	sí
n3g1b2	0.9437	0.0080	no
n3g1b3	0.9638	0.0965	sí
n3g2b1	0.8506	$3,19 \times 10^{-6}$	no
n3g2b2	0.9550	0.0270	no
n3g2b3	0.9493	0.0169	no
n3g3b1	0.6831	$4,26 \times 10^{-10}$	no
n3g3b2	0.6862	$4,85 \times 10^{-10}$	no
n3g3b3	0.9273	0.0043	no

Cuadro 3: Información individual de los datos.

Carga	Participantes	Promedio	Desv. Std.	Varianza	Mediana	Rango Intercuartil
n1g1b1	60	0.3529	0.1658	0.0275	0.3500	0.2205
n1g1b2	60	0.3477	0.1456	0.0212	0.3269	0.1907
n1g1b3	60	0.1381	0.0695	0.0048	0.1290	0.0914
n1g2b1	60	0.3646	0.1690	0.0285	0.3428	0.2715
n1g2b2	60	0.3638	0.1509	0.0227	0.3661	0.2057
n1g2b3	60	0.1362	0.0800	0.0064	0.1247	0.1055
n1g3b1	60	0.4617	0.1537	0.0236	0.4796	0.2035
n1g3b2	60	0.3875	0.1923	0.0370	0.3921	0.2635
n1g3b3	60	0.1261	0.0733	0.0053	0.1057	0.1044
n2g1b1	60	0.5050	0.1501	0.0225	0.5045	0.1790
n2g1b2	60	0.5195	0.1621	0.0262	0.5080	0.2214
n2g1b3	60	0.1777	0.0935	0.0087	0.1714	0.1429
n2g2b1	60	0.4951	0.1835	0.0337	0.5555	0.2602
n2g2b2	60	0.5209	0.1671	0.0279	0.5348	0.2457
n2g2b3	60	0.1741	0.1011	0.0102	0.1754	0.1458
n2g3b1	60	0.5597	0.1499	0.0224	0.6032	0.1620
n2g3b2	60	0.5729	0.1681	0.0282	0.6300	0.1871
n2g3b3	60	0.1428	0.0767	0.0058	0.1333	0.1105
n3g1b1	60	0.7728	0.1210	0.0146	0.7726	0.1828
n3g1b2	60	0.7600	0.1213	0.0147	0.7769	0.2076
n3g1b3	60	0.2125	0.1066	0.0113	0.2058	0.1470
n3g2b1	60	0.7751	0.1657	0.0274	0.8229	0.1574
n3g2b2	60	0.7377	0.1573	0.0247	0.7642	0.2066
n3g2b3	60	0.1835	0.1030	0.0106	0.1746	0.1406
n3g3b1	60	0.8546	0.1717	0.0295	0.9110	0.1005
n3g3b2	60	0.8348	0.1612	0.0260	0.8816	0.1084
n3g3b3	60	0.2101	0.1391	0.0193	0.1819	0.2083

Cuadro 4: Resultados obtenidos de prueba Kruskal-Wallis.

Chi cuadrada	DF	P
1123	26	$2,2 \times 10^{-16}$

4. Conclusión

Con la información que nos arroja la figura 1, podemos concluir que el valor de **F-score** depende de la probabilidad con la que aparezcan los colores.

A mayor o mejores probabilidades en conjunto, el **F-Score** será mayor. Considerando que contamos con 3 probabilidades para cada color, tenemos un total de 27 combinaciones de probabilidades. mismas que están ordenadas de mayor a menor. Para cada grupo de tres cajas, la primer caja representa la combinacion de todas las probabilidades mas altas y, la última caja la peor combinación.

5. Reto 1

Utilicé una página de internet [3] para ver cuáles eran los caracteres **ASCII** que iba a utilizar. Para darme una idea de como los iba a dibujar, utilice una hoja de excel para visualizarlos. En un archivo `.txt` escribí la secuencia que correspondería para cada simbolo, el archivo `.txt` se encuentra en mi repositorio [4]. Utilicé los dígitos del 0 al 9, agregando además los caracteres del cuadro 5

Cuadro 5: Caracteres utilizados.

¿	?	\$	¥
!	i	±	
{	}	ï	ë

Utilicé como base uno de los códigos vistos en clase [5] al cual le apliqué unos pequeños cambios.

Listing 6: Segmento de código reto 1.

```

r <- 7
c <- 5
dim <- r * c

tasa <- 0.15
tranqui <- 0.99

tope <- 21
digitos <- 0:tope
k <- length(digitos)
contadores <- matrix(rep(0, k*(k+1)), nrow=k, ncol=(k+1))
rownames(contadores) <- 0:tope
colnames(contadores) <- c(0:tope, NA)

```

Listing 7: Matriz de confusión.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	<NA>
0	4	0	12	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	14	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
5	0	4	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	4	4	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	11	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	1	0	0	0	10	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	0	9	0	0	0	1	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	3	8	0	0	0	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	12	2	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	2	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	3
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	1	0
17	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
18	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	12	0	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	15	0

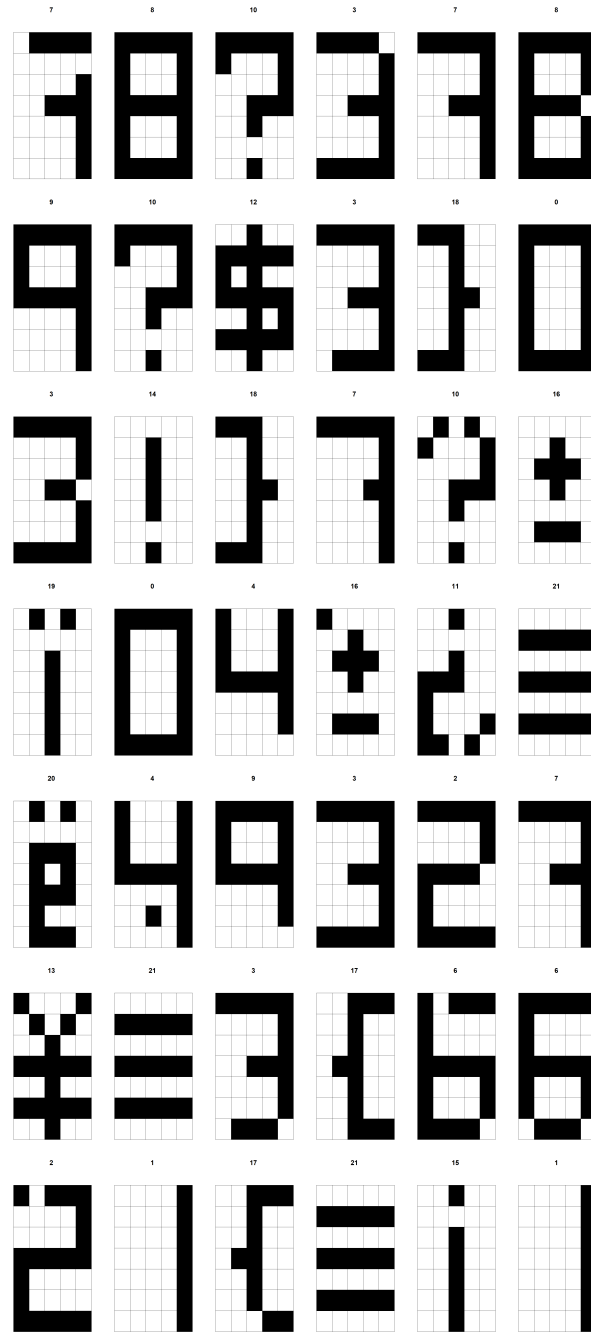


Figura 2: Imágenes obtenidas.

Referencias

- [1] Said Bleik. Computing classification evaluation metrics in r., 2016. URL https://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html.
- [2] Jason Brownlee. How to calculate precision, recall, and f-measure for imbalanced classification, 2020. URL <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>.
- [3] El codigo ASCII. El codigo ascii., 2020. URL <https://elcodigoascii.com.ar/>.

- [4] Claudia Hernández. archivo txt., 2021. URL <https://github.com/claudiahr/Simulacion-AD2021/tree/main/HW12>.
- [5] Elisa Schaeffer. Código base, 2021. URL <https://github.com/satuelisa/Simulation/blob/master/NeuralNetwork/ann.R>.